



# Proceeding

## Seminar Nasional Riset Teknologi Informasi 2013

**Social Informatics:  
The Social Consequences, the Applications,  
and the Use of ICT Tools**

Yogyakarta, 31 Agustus 2013

Aplikasi  
Algoritma  
Basis Data  
Multimedia  
Sistem Cerdas  
Perangkat Keras  
Jaringan Komputer  
Pengolahan Citra dan Grafika  
Sosial dan Informatika Sosial  
Sistem Pendukung Keputusan

Diselenggarakan Oleh :



YAYASAN PENDIDIKAN WIDYA BAKTI  
**STMIK  
AKAKOM**  
YOGYAKARTA  
Yang Pertama dan Utama

Parsing Data MySQL ke File XML untuk Pencarian Data <i>Thomas Edyson Tarigan</i> .....	109
Analisa dan Perancangan e-CRM untuk Mendukung Strategi Bisnis di SoloNet Internet Service Provider <i>Widyo Ari Utomo</i> .....	117
<b>D. Jaringan Komputer</b>	
Pemanfaatan Protokol Group Signature untuk Alternatif Pengamanan pada Aplikasi iPowerMeeting <i>Amiruddin</i> .....	135
Implementasi RemoteApp untuk Private Cloud Computing pada Perusahaan Farmasi dengan Pendekatan Infrastructure as a Services (IaaS) <i>Muhammad Noval Riswandha</i> .....	141
Analisis Tabrakan Data pada Jaringan Ad-Hoc Multinode untuk Sistem Komunikasi Kapal Laut <i>Mukminatun Ardaizi</i> .....	151
Analisis Perencanaan Coverage Area WiFi 802.11g di dalam Pesawat Udara Pesawat Boeing 737-900ER <i>Puji Edriany Santoso, Uke Kurniawan Usman, Tengku A. Riza</i> .....	157
Analisis Hubungan Kausalitas antara Konsumsi Daya Listrik dan Trafik Internet Spasial Kampus <i>Sis Soesetijo</i> .....	165
Rancang Bangun Wireless Sensor Network untuk Monitoring Pencemaran Udara <i>Syahrir</i> .....	171
<b>E. Multimedia</b>	
Virtual Web 3D untuk Garment Modelling Berdasarkan Anthropometry <i>Endra Rahmawati</i> .....	183
<b>F. Pengolahan Citra dan Grafika</b>	
Komparasi Teknik Akselerasi untuk Representasi Online Menggunakan Gaya Visualisasi Virtual Reality <i>Mursid W. Hananto</i> .....	193
<b>G. Perangkat Keras</b>	
Perancangan Simulink Model Dari AR.Drone Sebagai Simulator Kontrol Quadrotor <i>Agung Prayitno</i> .....	203
Mp3 Player Portable Untuk File Talking Book Bagi Tuna Netra <i>Andrew Joewono, ST, MT., Diana L. Antonia, ST, MT., Steven Anthonius</i> .....	211
Self Stabilizing 1 Axis QuadCopter Using T2-Fuzzy Controller <i>Hendi Wicaksono</i> .....	219
Pemodelan ARIMA untuk Redaman Kanal HF Link Banda Aceh-Surabaya <i>Indra Jaya, Achmad Mauludiyanto</i> .....	227
Pengaruh Perbedaan Bentuk dan Lebar Slot Aperture Pada Antena Mikrostrip Aperture Coupled <i>Ipan Suandi</i> .....	233
Event Driven Framework Untuk Pengembangan Firmware pada Mikrokontroler <i>Listiarso Wastuargo, Inggriani Liem, Achmad Imam Kistijantoro</i> .....	239
Perancangan Light Follower Robot Menggunakan Sensor LDR dan Handphone Sebagai Pengendali Berbasis Mikrokontroler AT 89S51/52 <i>Lukman Hakim</i> .....	247
Modulator dan demodulator BPSK pada Platform Software Defined Radio dengan TMS320C6416 <i>Nicodemus FR Hutabarat, Achmad Affandi</i> .....	253
Analisa Efek Doppler pada Bandwidth Radio untuk Aplikasi Telemetry Data Roket <i>Sri Kliwati</i> .....	259
Desain Sistem Pemanenan Energi Gelombang Elektromagnetik pada Frekuensi GSM 900 <i>Widya Cahyadi</i> .....	263

# PARSING DATA MYSQL KE FILE XML UNTUK PENCARIAN DATA

Thomas Edyson Tarigan,<sup>1)</sup>

<sup>1)</sup>*Jurusan Teknik Informatika, STMIK AKAKOM Yogyakarta  
Jl. Raya Janti 143, Karangjambe Yogyakarta  
081568433388  
E-mail : tarigan@akakom.ac.id*

## Abstrak

Datasebagai sumber untuk menghasilkan informasi yang bermanfaat bagi user, dengan format data yang sudah baku untuk proses pengembangan sistem informasi akan menimbulkan masalah. Dengan melakukan parsing data ke dalam bentuk file XML maka data tersebut akan memiliki nilai tambah dalam pemanfaatannya untuk menghasilkan informasi.

Model iterative testing digunakan sebagai model untuk pengembangan file XML, pencarian data adalah salah satu kasus yang digunakan untuk memanfaatkan hasil parsing dari format Mysql ke format XML, dengan dilakukannya parsing ini maka pengembangan informasi multiplatform akan dapat ditangani dengan lebih baik.

Proses yang dilakukan adalah membuat penyimpanan data pada sebuah database Mysql kemudian data yang berada pada database tersebut di parsing ke format file XML, dalam proses parsing yang dibutuhkan class *DOMDocument* berfungsi sebagai sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. *DOM* menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk merubah isian presentasi visual. Pada dasarnya, *DOM* menghubungkan halaman web dengan script atau bahasa pemrograman.

Keyword : *Parsing, XML, Iterative Testing, DOM Document,*

## 1. Pendahuluan

Peningkatan secara kuantitatif dan kualitatif pengguna komputer sejalan dengan waktu maka secara angka statistik pertumbuhannya semakin meningkat, peningkatan ini juga diimbangi dengan meningkatnya teknologi dalam penggunaan software dan hardware. Khususnya perkembangan aplikasi yang berhubungan dengan website, mengingat semua lapisan masyarakat saat ini semakin mudah mendapatkan informasi pada dunia maya, maka peningkatan pengguna website terus mengalami peningkatan yang signifikan.

Seiring dengan peningkatan penggunaan website yang bertujuan dalam mencari informasi pada database, maka salah satu cara yang memungkinkan untuk diimplementasikan adalah dengan melakukan parsing data yang berada pada database berbasis Mysql ke dalam bentuk XML.

*parsing* adalah suatu cara memecah-mecah suatu rangkaian masukan (misalnya

dari berkas atau *keyboard*) yang akan menghasilkan suatu pohon uraian (*parse tree*) yang akan digunakan pada tahap kompilasi berikutnya yaitu analisis semantik.

Permasalahan yang menjadi pokok bahasan adalah bagaimana cara memecah – mecah suatu rangkain inputan dan akan menghasilkan *tree* dengan format data XML. Hasil dari *tree* dengan metode *iterative testing* dapat diolah untuk menghasilkan informasi, salah satunya dapat digunakan untuk melakukan pencarian informasi yang berasal dari *tree* tersebut.

Untuk itu dibutuhkan suatu input yang mampu menyimpan data tersebut kedalam suatu database, data yang berasal dari database tersebut diparsing ke dalam bentuk *tree*. Data yang awalnya tersimpan dalam database berbasis mysql sudah diubah dalam format XML.

Pada penelitian Iacob *et al* [2004] membahas arsitektur XML parser untuk XML paralel. Parser ini menggunakan struktur data GODDAG dengan

menempatkan DOM *Tree* tradisional untuk menyimpan markup secara paralel pada isi dokumen dan memanfaatkan DOM API sebagai alat yang memungkinkan bagi *software developers* bersamaan dengan dokumen XML untuk digunakan, bukan parsing setiap komponen individu dengan tradisional DOM XML parser. Penelitiannya ini juga menggambarkan arsitektur dari parser, struktur data dan algoritma yang digunakan dan seperti DOM API.

Adapun tujuan dari penelitian ini adalah mengimplementasikan proses parsing data dari format Mysql ke dalam format XML sehingga data yang dihasilkan dapat dimanfaatkan untuk *multiplatform*. Dengan memanfaatkan model *iterative testing* maka akan memberikan keleluasaan dalam menghasilkan informasi yang dibutuhkan pengguna sistem.

## 2. Landasan Teori

### 2.1 XML

XML adalah sebuah versi penyederhanaan dari *Standard Generalized Markup Language* (SGML), sebuah sistem untuk mengatur dan menandai elemen-elemen dari sebuah dokumen. SGML dikembangkan dan distandarisasi oleh *International Organization for Standards* (ISO) pada tahun 1986. SGML masih digunakan secara luas di dalam industri penerbitan, namun karena ukurannya yang besar dan kerumitannya, SGML menjadi kurang ideal untuk penggunaan secara luas di era Internet. Untuk memecahkan masalah-masalah ini, *World-Wide Web Consortium* (W3C) membuat XML sebagai sebuah *subset* dari SGML. Walaupun kini XML mempunyai fitur-fitur yang tidak didukung SGML, sehingga XML tidak lagi merupakan sebuah subset.

Seperti dijelaskan oleh Marchal[1] dalam bukunya, XML awalnya dirancang sebagai HTML yang lebih baik. Sejalan dengan perkembangan popularitasnya, kegunaan XML sekarang menjadi lebih luas ketimbang tujuan awalnya. Nampak bahwa XML telah menjadi sebuah standar untuk pertukaran data, di mana penggunaannya tidak hanya untuk halaman *web* saja, namun juga untuk data yang lain. Melalui berbagai inisiatif seperti *BizTalk*, *RosettaNET* dan *ebXML* yang bertujuan untuk memajukan penggunaan XML, memungkinkan perniagaan elektronik dan melakukan penggabungan aplikasi – serta standar seperti *Simple Object Access Protocol* (SOAP)[2] yang merupakan protokol untuk pertukaran informasi di sebuah lingkungan terdesentralisir dan terdistributif. Perkembangan XML telah melampaui kegunaan

awalnya. Pada saat standar integrasi dan pertukaran data telah gagal, atau setidaknya tidak dapat mencapai potensinya secara penuh, XML akan sangat mungkin menjadi sukses.

Ciri kunci dari XML adalah sifat keterbukaannya. Seperti yang ditulis dalam *Oxford Dictionary of Computing*[3], *open system* didefinisikan sebagai berikut: “*Any system in which the components conform to non-proprietary standards rather than to the standards of specific supplier of hardware or software*”. Karena standar XML serta berbagai standar pendukungnya didefinisikan oleh sebuah konsorsium (W3C), maka XML memenuhi kriteria *open system*.

#### A. Konsep XML Secara Luas

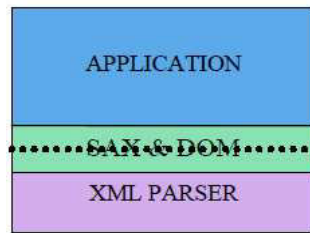
Walaupun sebuah keberadaan dokumen XML hanya dapat menampung teks yang diberi *markup* dengan berbagai *tag* seperti telah dibahas sebelumnya, sebuah dokumen XML juga dapat mengandung hal-hal lain. Nyatanya ada tiga kesatuan yang dapat diuraikan oleh sebuah aplikasi XML[4]:

- *XML Document* Mengandung data dokumen, dietikatkan dengan unsur-unsur XML yang mempunyai arti, beberapa di antaranya dapat mengandung perlengkapan.
- *Document Type Definition* (DTD) Menentukan aturan-aturan bagaimana unsur-unsur dan perlengkapan perlengkapan, dan penempatan data lainnya dari sebuah dokumen yang tunduk pada standar XML.
- *Stylesheet* Memerintah bagaimana sebuah dokumen sebaiknya dibentuk ketika mereka ditampilkan.

Berbagai *stylesheet* yang berbeda dapat diterapkan pada dokumen yang sama, sehingga mengubah penampilannya tanpa mempengaruhi data yang mendasarinya. Maka dari itu, DTD menentukan informasi apa saja yang dapat disimpan di dalam sebuah berkas XML dan *stylesheet* mengubah bentuk satu dokumen XML menjadi bentuk yang lain. DTD dan *stylesheet* ditetapkan di dalam standarnya masing-masing, yang akan ditinjau pada dua bagian berikut ini.

#### B. Pengurai XML (XML Parser)

Walaupun struktur dokumen XML termasuk sederhana, pembuatan program untuk menguraikan dokumen-dokumen XML tidak dapat dikatakan suatu tugas yang sederhana. Oleh karena itu muncul suatu jenis perangkat lunak yang disebut Pengurai XML (*XML Parsers*).



Gambar 1 Arsitektur Pengurai XML Secara Umum

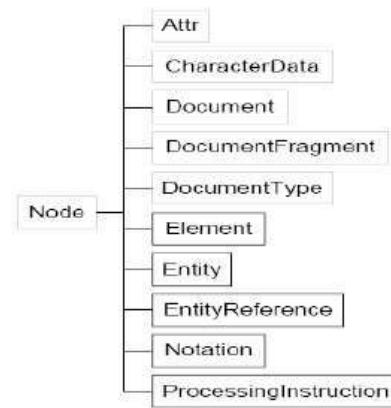
Sebuah pengurai XML umumnya berbentuk pustaka (*software library*) tingkat menengah yang memberikan layanan-layanan bagi aplikasi pada tingkat yang lebih tinggi untuk membaca serta mengambil data yang terletak di dalam dokumen XML. Pengurai-pengurai XML ini menetapkan API tertentu untuk berinteraksi dengan program aplikasi yang menggunakannya. API ini mendefinisikan *data model* dari sebuah dokumen XML kepada aplikasi yang menggunakan pengurai tersebut.

Terdapat dua standar API untuk kegunaan ini: *Simple API for XML* (SAX) dan *Document Object Model* (DOM). SAX tercipta awalnya melalui proses informal yang terjadi di *mailing list* XML-DEV kemudian diterbitkan oleh David Megginson di <http://www.megginson.com/SAX>[5]. Sedangkan DOM ditetapkan oleh W3C – sebuah badan formal yang mengawasi dan memberikan acuan-acuan untuk perkembangan *world-wide web*. Karena menggunakan paradigma yang berbeda, kedua API ini tidak menjadi saingan satu sama lain, melainkan saling melengkapi. Tidak jarang pustaka-pustaka pengurai XML memberikan kedua pilihan API bagi program-program aplikasi.

### C. Simple API for XML (SAX)

Pustaka pengurai XML menggunakan fungsi-fungsi ataupun antarmuka *callback* untuk memanggil aplikasi pada saat dokumen XML diuraikan. Pusat kendali terletak pada pustaka pengurai ketimbang program aplikasi. Penggunaan SAX lebih cocok pada aplikasi-aplikasi yang menggunakan XML sebagai representasi data eksternal dan menggunakan struktur datanya sendiri yang dapat berbeda jauh dari struktur dokumen-dokumen XML yang digunakannya. Sejalan dengan penguraian XML, fungsi-fungsi *callback* yang dipanggil pengurai XML mengisi struktur data aplikasi berdasarkan parameter-parameter yang diberikan.

Karena pengurai XML tidak perlu menyimpan data pada saat dokumen diuraikan, aplikasi-aplikasi yang menggunakan SAX cenderung memerlukan lebih sedikit *memory* untuk memproses dokumen XML. Selain itu, sifat penguraian SAX



Gambar 2 Hirarki Tipe Dalam DOM

Dalam DOM, dokumen XML memiliki tipe *Document*. Elemen-elemen di dalam dokumen tersebut umumnya bertipe *Element*. Berbagai atribut yang dimiliki oleh elemen diwakili oleh obyek-obyek bertipe *Attr*. Data tekstual seperti komentar dan elemen yang berisi teks diwakili oleh *CharacterData*. Sedangkan ada beberapa tipe khusus lainnya seperti *Entity*, *EntityReference*, *Notation*, dan *ProcessingInstruction*. Kesemua tipe ini merupakan turunan dari tipe *Node*. Elemen terluar atau *root node* dari suatu dokumen bukanlah juga merupakan dokumen itu sendiri. Sebuah keberadaan dari *Document* memiliki paling banyak satu keberadaan dari *Node* yang menjadi *root node*. Sedangkan *Document* yang tidak mempunyai *root node* dikatakan dokumen kosong (*blank document*).

W3C mendefinisikan API DOM dalam bentuk *Interface Definition Language* (IDL) – sebuah standar yang digunakan oleh *Object Management Group* (OMG) untuk mendefinisikan antarmuka berorientasi obyek untuk komponen-komponen perangkat lunak namun tidak bergantung pada suatu bahasa pemrograman tertentu. Didefinisikannya DOM

dalam IDL memungkinkan penerapan XML Parser untuk berbagai bahasa pemrograman – terdapat XML parser untuk C++, Java, SmallTalk, Ada, bahkan COBOL.

## 2.2 Database

Pengertian sistem *database* menurut Peter Rob dan Carlos Coronel[6] adalah: “*A database is a shared, integrated computer structure that houses the collection of end user data, that is, raw facts of interest to the end user, metadata, or data about data, through which the data are integrated*”.

Menurut Kok Yung, basis data dapat diartikan sebagai kumpulan data yang terdiri atas

satu atau lebih tabel yang terintegrasi satu sama lain, dimana setiap pemakai diberi wewenang untuk dapat mengakses (mengubah, menghapus, menganalisis, menambah, memperbaiki) data dalam tabel-tabel tersebut.

DBMS adalah perangkat lunak komputer yang digunakan untuk menciptakan, mengakses, mengontrol, dan mengatur basis data. DBMS dapat dibentuk dari komponen-komponen:

- a. Data yang disimpan dalam basis data, mencakup data numerik dan non-numerik yang terdiri dari karakter, waktu (tanggal dan jam), logika, dan data lain yang lebih kompleks seperti gambar dan suara.
- b. Operasi standar yang disediakan oleh hampir semua DBMS, yang melengkapi pengguna dengan kemampuan dasar untuk memanipulasi basis data.
- c. *Data Definition Language* (DDL) merupakan bahasa yang digunakan untuk mendeskripsikan isi dan struktur basis data. DDL dapat digunakan untuk mendeskripsikan nama-nama atribut, tipe data, serta lokasi di dalam basis data.
- d. *Data Manipulation Language* (DML) atau bahasa *query* ini membentuk perintah-perintah untuk masukan, keluaran, *editing*, analisis basis data. DML yang telah distandarisasikan disebut *Structured Query Language* (SQL).
- e. Bahasa pemrograman (*programming tools*). Di samping perintah-perintah dan *queries*, basis data juga harus dapat diakses secara langsung oleh program-program aplikasi melalui *function calls* (atau *subroutine calls*) yang dimiliki oleh bahasa-bahasa pemrograman konvensional.
- f. Struktur *file*. Setiap DBMS memiliki struktur internal yang digunakan untuk mengorganisasikan data walaupun beberapa model data yang umum telah digunakan oleh sebagian besar DBMS.

### 2.3 Metode Iterative Testing

Iteratif desain adalah metodologi desain berdasarkan proses siklus prototyping, menguji, menganalisa, dan memperbaiki pekerjaan yang sedang berjalan. Dalam desain iteratif, interaksi dengan sistem yang dirancang digunakan sebagai bentuk penelitian untuk menginformasikan dan berkembang proyek, sebagai versi yang berurutan, atau iterasi dari desain diimplementasikan.[8]

Pengembangan dengan iterative merupakan sebuah pendekatan pada pengembangan software yang lebih memusatkan gagasan pengembangan dalam bentuk siklus, dari pada melakukan semuanya secara sekaligus. Metode ini tidak sesuai pada semua

jenis software namun dapat sangat bermanfaat dan sangat menguntungkan pada pengaturan tertentu. Merupakan hal yang penting untuk dipahami, bahwa pengembangan iterative bukanlah merupakan sebuah pengembangan yang tidak terencana atau spontan. Faktanya, proses pengembangan software terstrukturisasi dengan baik.

Pada pengembangan iterative, memulainya dengan tahapan perencanaan project, kemudian bergerak pada tahapan pengembangan lainnya dan pada akhirnya merilis produk.

Sejalan dengan rilis produk, terdapat hasil yang diperoleh dari pengujian produk (*product testing*) dan pengujian pada pengguna (*user testing*). Hasil tersebut akan digabungkan dengan rilis selanjutnya. Dimana rilis software pada pengembangan *iterative* dapat berupa rilis software di tahapan awal pengembangan, bukan rilis produk pada publik.[7]

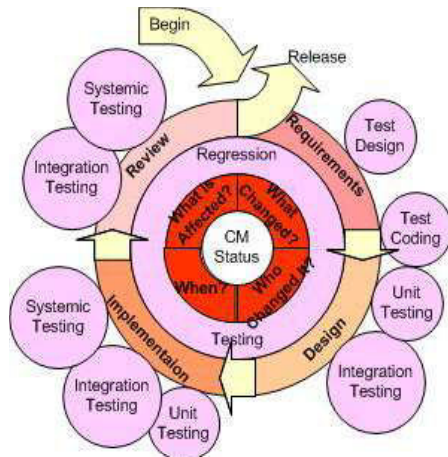
Pengujian merupakan teknik yang digunakan untuk memvalidasi proses, harus didistribusikan dan dilakukan pada setiap tahap dari siklus pengembangan perangkat lunak.

Sistem pengembangan dan desain uji harus dilakukan secara bersamaan. uji desain harus disertakan disetiap tahapan siklus hidup, dan pengukuran harus terjadi pada setiap akhir fase.

Pengembangan proyek dimulai dengan menspesifikasikan dan mengimplementasikan arsitektur / prototype dari perangkat lunak yang telah dirancang sebelumnya.

Komponen perangkat lunak diciptakan dari kebutuhan bisnis yang diidentifikasi, dan kemudian mengidentifikasi persyaratan lebih lanjut. Proses ini kemudian diulang, sampai menghasilkan rancangan terbaik dari perangkat lunak untuk setiap siklus dari model.

Pada iterative model pengujian yang dilakukan dengan langkah sebagai berikut :[8]



Gambar 3 Dasar Model Iterative Testing

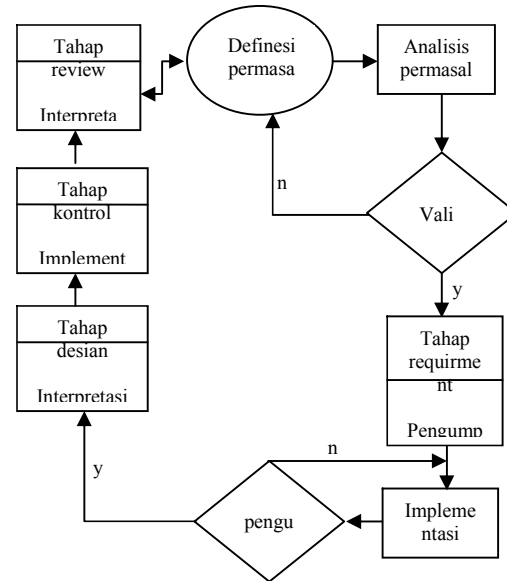
Empat proses dasar dari model iteratif adalah:

- Tahap requirement, kebutuhan untuk perangkat lunak dikumpulkan dan dianalisis. Iterasi akhirnya harus menghasilkan spesifikasi yang lengkap.
- Tahap desain, arsitektur perangkat lunak memenuhi persyaratan yang dirancang, dalam hal ini dapat berupa desain baru, atau perbaikan dari desain sebelumnya.
- Tahap implementasi, ketika software ini telah selesai pada phase coding, telah terintegrasi dan telah diuji.
- Tahap review, perangkat lunak dievaluasi, requirement di review, perubahan dan penambahan requirement pada tahap ini dapat dilakukan.

### 3. Hasil

#### Rancangan Sistem

Penelitian ini menggunakan metode *iterative testing* untuk perancangan sistem pada tahap requirement untuk kasus parsing database ke dalam format xml dengan pencarian data karyawan dapat digambarkan seperti berikut,



Gambar 4 Skema Analisis Sistem

Dalam penenerapan *model iterative testing* tahap yang pertama adalah tahap *requirement* yaitu proses pengumpulan informasi dan analisis informasi mengenai data – data karyawan yang nantinya akan dirancang database-nya. Hasil dari analisis ini digunakan untuk mengidentifikasi permintaan organisasi terhadap sistem yang akan dirancang.

Untuk memperoleh informasi yang dibutuhkan tersebut, maka diperlukan langkah *fact finding*, yang bertujuan untuk mencari informasi dan fakta mengenai karyawan untuk itu dibutuhkan suatu penelitian yang bertujuan untuk memahami situasi. Misalnya data apa saja yang diinginkan serta penggunaan sistem nantinya. Pada tahap *fact finding* cara yang dilakukan dengan melihat data yang ada dilaporkan tentang informasi karyawan, wawancara langsung dengan karyawan bersangkutan. Dalam proses *fact finding* ini langsung melibatkan pengguna sistem, tujuannya agar tidak terjadi bias dalam memahami situasi yang ada.

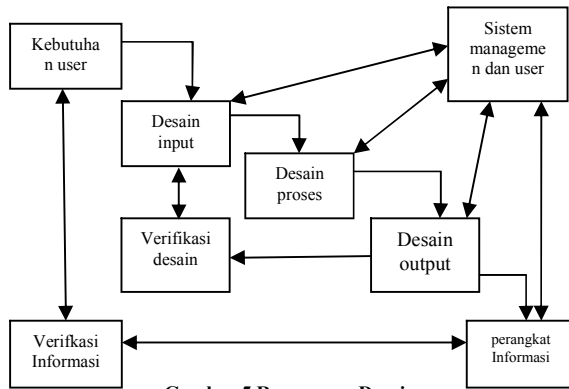
Dari tahap *fact finding* akan menghasilkan *requirement specification document*. Adapun dokumen yang dihasilkan adalah dokumen yang berisi data dan informasi yang dibutuhkan oleh sistem saat ini. Misalnya data yang disimpan pada tiap karyawan adalah id karyawan, nama lengkap, jenis kelamin, alamat, tanggal lahir dan nomer telpn.

Selanjutnya dilakukan proses mengidentifikasi kebutuhan sistem yang akan didukung oleh database untuk melakukan proses transaksi yaitu, entry data, update atau delete dan query data.



## Rancangan Desain

Rancangan desain dimulai dari kebutuhan user sesuai dengan permasalahan yang sudah didefinisikan. Rancangan yang dimulai dari rancangan input, proses dan output, adapun rancangannya seperti dibawah ini,



Gambar 5 Rancangan Desain

Sesuai dengan model *iterative testing* setiap tahapan dalam proses perancangan selalu melibatkan user/pengguna sistem dan manajemen sehingga dalam proses implementasi sistem akan cenderung mengalami perubahan desain yang terus berkembang untuk penyempurnaan sistem nantinya.

## Implementasi

Pengisian data ke dalam database Mysql dilakukan dengan mengisi data dari form. Adapun hasil dari pengisian data ke dalam tabel Mysql adalah seperti dibawah ini,

Data Karyawan					
NIP	NAMA	ALAMAT	NO. TELPON	JABATAN	Aksi
110	Adi Setiawan	J. Sumbada 417	081534432751	Manager Produksi	Aksi / Edit / Delete
111	Olga Sukarya	J. Mangku 14	08129367754	Administrasi Umum	Aksi / Edit / Delete
112	Joko Silawanto	J. Marguhutani 275	08125434111	Sales	Aksi / Edit / Delete
113	Andi Kurnia	J. Depok 33 Siman	08164613999	Manager Pemasaran	Aksi / Edit / Delete
114	Panda	J. Sumatera 1234	08144646999	Office Boy	Aksi / Edit / Delete
115	Ruli Susanti	J. Raya Janti 21	0899999999999	Diraktor	Aksi / Edit / Delete
116	Samet	J. Serrigin 111	0816666666666	Televisi	Aksi / Edit / Delete
117	Agus Aca	J. Pacitan 1067	99999999	adrese	Aksi / Edit / Delete
118	Willy	J. Sengapat 124	08122251234	HRD	Aksi / Edit / Delete
119	Billy	J. Darisan 23	0811122222	Manager Area	Aksi / Edit / Delete
120	Susanto Wijanarko	J. Selatan Raya 4789	0813223134444	Kasaman Kantor	Aksi / Edit / Delete
121	xxxxxxxx	xxxxxxxxxx	88888888	4000000000000	Aksi / Edit / Delete
123	scwerfGj	hjkjklmnop	3475757575	tbl_nm_jtgh	Aksi / Edit / Delete

Gambar 6 Pengisian Data ke Database Mysql

Proses ini dilakukan oleh admin, pada halaman form ini admin dapat melakukan proses CRUD.

## Proses Parsing ke Format XML

Proses parsing dilakukan dengan melakukan koneksi ke database dan nama table yang digunakan di Mysql kemudian mengambil data yang ada pada database Mysql kemudian data tersebut hasilnya

disimpan dalam sebuah array, selanjutnya data array di konversi dalam format XML.

Untuk itu dibutuhkan class *DOMDocument* berfungsi sebagai sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. DOM menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk merubah isidan presentasi visual. Pada dasarnya, DOM menghubungkan halaman web dengan script atau bahasa pemrograman. Untuk membentuk elemen baru pada data XML, dimana pegawai sebagai root dari data file XML, sedangkan *\$dom->appendChild(\$pegawai)*; yang membentuk elemen baru sebagai child dari file XML tersebut. Selanjutnya membuat text baru pada node XML yang datanya diambil dari \$q, setelah itu melakukan penambahan sebuah node diakhir objek yang telah ditentukan.

Setelah proses konversi ke dalam format XML dilakukan maka dilakukan proses penyimpanan data dalam format XML. Adapun proses tersebut dilakukan dengan skrip seperti dibawah ini,

```
mysql_connect('localhost','root','root');
mysql_select_db('db_karyawan');
$sql = "select * from `tbl_karyawan`";
$q = mysql_query($sql) or die(mysql_error());

$dom = new DOMDocument();

$pegawai = $dom->createElement('pegawai');
$dom->appendChild($pegawai);

while($r = mysql_fetch_array($q)) {

    $nip = $dom->createElement('nip');
    $nipText = $dom->createTextNode($r['nip']);
    $nip->appendChild($nipText);

    $nama = $dom->createElement('nama');
    $namaText = $dom->createTextNode($r['nama']);
    $nama->appendChild($namaText);

    $alamat = $dom->createElement('alamat');
    $alamatText = $dom->createTextNode($r['alamat']);
    $alamat->appendChild($alamatText);

    $no_hp = $dom->createElement('no_hp');
    $no_hpText = $dom->createTextNode($r['no_hp']);
    $no_hp->appendChild($no_hpText);

    $jabatan = $dom->createElement('jabatan');
    $jabatanText = $dom->createTextNode($r['jabatan']);
    $jabatan->appendChild($jabatanText);

    $tbl_karyawan = $dom->createElement('tbl_karyawan');
    $tbl_karyawan->appendChild($nip);
    $tbl_karyawan->appendChild($nama);
    $tbl_karyawan->appendChild($alamat);
    $tbl_karyawan->appendChild($no_hp);
    $tbl_karyawan->appendChild($jabatan);

    $pegawai->appendChild($tbl_karyawan);
}

$xmlString = $dom->saveXML();
echo $xmlString;
?>
```

Gambar 7 Skrip Proses Ubah ke Array



Setelah proses penyimpanan dapat dilakukan maka data dalam bentuk format XML seharusnya sudah dalam bentuk struktur XML, adapun struktur XML tersebut seperti dibawah ini,

```
<root>
<child>
<subchild>.....</subchild>
</child>
</root>
```

Adapun hasil dari proses parsing Mysql ke dalam format XML tersebut seperti dibawah ini,



```

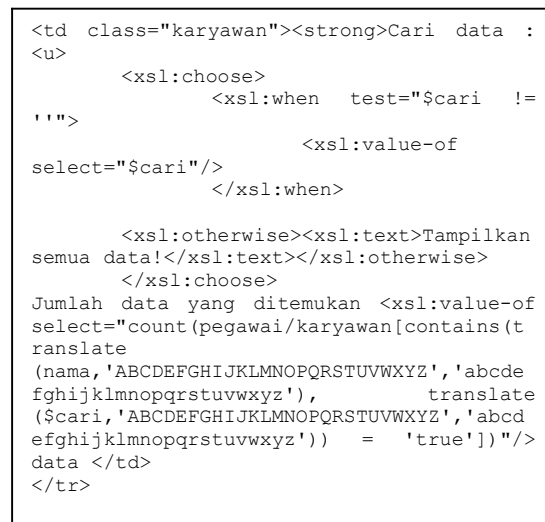
<karyawan>
  <nip>111</nip>
  <nama>Gita Sukaryo</nama>
  <alamat>Jl.Nangka 24</alamat>
  <no_hp>081255367754</no_hp>
  <jabatan>Administarsi Umum</jabatan>
</karyawan>
<karyawan>
  <nip>112</nip>
  <nama>Joko Siswanto</nama>
  <alamat>Jl.Mangkubumi 235</alamat>
  <no_hp>08325434111</no_hp>
  <jabatan>Sales </jabatan>
</karyawan>

```

**Gambar 8 Hasil Parsing Mysql ke Format XML**

Dalam penelitian ini salah satu kasus untuk memanfaatkan data format XML tersebut adalah melakukan proses pencarian data pada database yang berbasis XML. Dimana data awal berasal dari data yang tersimpan dalam format database Mysql.

Adapun proses untuk melakukan pencarian data dari format data XML adalah seperti dibawah ini,



```

<td class="karyawan"><strong>Cari data :
<u>
  <xsl:choose>
    <xsl:when test="$cari !=
    ''">
      <xsl:value-of
      select="$cari"/>
    </xsl:when>

    <xsl:otherwise><xsl:text>Tampilkan
    semua data!</xsl:text></xsl:otherwise>
  </xsl:choose>

  Jumlah data yang ditemukan <xsl:value-of
  select="count(pegawai/karyawan[contains(t
ranslate
(nama,'ABCDEFGHJKLMNOPQRSTUVWXYZ','abcde
fghijklmnopqrstuvwxyz'), translate
($cari,'ABCDEFGHJKLMNOPQRSTUVWXYZ','abcd
efghijklmnopqrstuvwxyz')) = 'true']"/>
data </td>
</tr>

```

**Gambar 9 Skrip pencarian data dari format XML**

Skrip diatas merupakan proses untuk menampilkan kembali data yang dicari sesuai dengan

inputan yang telah diinputkan oleh user, untuk menampilkan data yang diinputkan tersebut dari file XML maka yang dilakukan adalah memformat data XML dengan menggunakan XSL (*eXtensible Stylesheet Language*), untuk memformat kode XML maka harus dibuat terlebih dahulu suatu stylesheet XSL. XML digunakan untuk mendefinisikan apa arti dari data dan bukan bagaimana data ditampilkan maka XML. Hasil Keluaran XSL bisa berupa HTML, Text file atau XML dengan format yang baru.

Dengan mengubah format database ke dalam bentuk format XML maka database menjadi lebih bersifat dinamis, artinya data tersebut dapat digunakan oleh bermacam – macam *tools* yang mendukung format XML. Bahkan data tersebut dapat dikembangkan ke basisdata dalam bentuk multiplatform.

Dengan menggunakan metode *iterative testing* pengembangan sistem yang bersifat jangka panjang yang memerlukan adaptasi cepat dan pengembangan terhadap perubahan sistem dapat dilakukan selama *tools* yang digunakan masih mendukung untuk penggunaan XML.

Untuk keberhasilan menggunakan model *iterative testing* pada pengujian sistem ini adalah mengutamakan keberhasilan sistem sebagai prioritas utama dengan menghasilkan produk yang bersifat berkelanjutan sesuai dengan perubahan yang diinginkan user, tidak adanya batasan waktu dalam penyelesaian sistem, adanya kerjasama yang terus menerus antara pengembang dengan pemilik sistem, arsitektur kebutuhan dan desain yang bagus muncul dari kebutuhan yang ada dilapangan sehingga membutuhkan evaluasi dan mencari cara untuk lebih efektif untuk menyelesaikan kasus.

## Kesimpulan

1. Lebih kecil kemungkinan untuk terjadi bug dalam sistem yg dirancang, ini disebabkan proses pengujian dilakukan pada setiap level tahapan dan tidak hanya melibatkan kelompok kerja, juga melibatkan kelompok yang tidak terkait langsung dengan proses pengembangan sistem. Sehingga komunikasi yang baik terjadi antara pengembang sistem dan pengguna sistem.
2. Pengembang sistem dapat bekerja lebih baik dalam menentukan kebutuhan pengguna sistem karena pengguna sistem berperan aktif dalam pengembangan sistem sehingga pengembang sistem mengetahui apa yang diharapkan pengguna sistem.

Sebaliknya kelemahannya adalah :

1. Proyek mungkin tidak akan pernah selesai karena akan lebih sering fokus pada perubahan prototipe, akibatnya dokumentasi, sistem backup dan recovery, sistem performance dan security akan lebih terlupakan sehingga berdampak terhadap pemeliharaan sistem menjadi lebih sulit.
2. Pengembang sistem kadang membuat implementasi yang sembarangan.

### Daftar Pustaka

Marchal, Benoit, *XML By Example*. Indianapolis, Que. 1999.  
World-Wide Web Consortium.. *Simple Object Access Protocol (SOAP) 1.1*.  
<http://www.w3c.org/>, 2000.  
*Oxford Dictionary of Computing* England, Oxford University Press, 1990.

Eckstein, Robert, *XML Pocket Reference*. California, O'Reilly, 1999.

<http://www.megginson.com/SAX>, Marc 1999.

Coronel, Rob, *Database System: Design, implementation and Management, 4<sup>th</sup> Edition*. Cambridge, Thomson Learning, 2000.

<http://www.snyders.us/qa-iterative.htm>

B. Campbell, Dr. G. Ray "Managing Iterative Testing in an Agile Development Project" Version 1.3 April 9, 2002.

### [CV Penulis]

Thomas E. Tarigan, menyelesaikan studi S1 bidang Teknik Informatika pada Universitas Kristen Duta Wacana. Pada tahun 2005 sampai saat ini merupakan staff edukatif di STMIK Akakom Yogyakarta.